AFHRL-TP-89-54

(2)

AIR FORCE

HUMAN RESOURCES

AD-A221 036

ACQUISITION OF PROGRAMMING SKILLS

Valerie J. Shute
Carmen M. Pena

MANPOWER AND PERSONNEL DIVISION
Brooks Air Force Base, Texas 78235-5601

April 1990
Final Technical Paper for Period August 1988 - August 1989

LABORATORY

AIR FORCE SYSTEMS COMMAND
BROOKS AIR FORCE BASE, TEXAS 78235-5601

NOTICE

WILLIAM E. ALLEY, Technical Director
Manpower and Personnel Division

DANIEL L. LEIGHTON, Colonel, USAF
Chief, Manpower and Personnel Division

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
April 1990

3. REPORT TYPE AND DATES COVERED
Final - August 1988 to August 1989

4. TITLE AND SUBTITLE
Acquisition of Programming Skills

5. FUNDING NUMBERS
PE - 61102F
PR - 2313
TA - T1
WU - 44

6. AUTHOR(S)
Valerie J. Shute
Carmen M. Pena

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Manpower and Personnel Division
Air Force Human Resources Laboratory
Brooks Air Force Base, Texas 78235-5601

8. PERFORMING ORGANIZATION REPORT NUMBER
AFHRL-TP-89-54

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*
The purpose of this research was to investigate the relationship between programming skill acquisition and various measures of individual differences including: (a) problem solving abilities (e.g., ability to decompose a problem into its constituent parts); (b) prior knowledge and general cognitive skills (e.g., arithmetic reasoning, word knowledge, information processing speed); and (c) passive versus active learning style. Ability measures consisted of scores from algebra tests assessing problem solving abilities on algebra word problems. The algebra test battery was constructed to parallel specific abilities hypothesized by Brooks (1977) to underlie programming skills: understanding, method-finding, and coding. Subjects (N = 265) received extensive Pascal programming instruction from an intelligent tutoring system. Following the instruction, and online battery of criterion tests was administered measuring the breadth and depth of programming knowledge and skills acquired from the tutor. Results indicated that successful learning of Pascal programming was a function of: certain problem solving abilities posited by Brooks (i.e., identifying/understanding a problem type and sequencing problem elements into a solution), working memory capacity (i.e., how much concurrent storage and processing an individual was capable of doing), and an active learning style. Implications of the findings for the development of a theoretical framework on which to base programming instruction are discussed.

14. SUBJECT TERMS
individual differences
problem solving ability
programming ability
skill acquisition

15. NUMBER OF PAGES
20

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

AFHRL Technical Paper 89-54 April 1990

ACQUISITION OF PROGRAMMING SKILLS

Valerie J. Shute
Carmen M. Pena

MANPOWER AND PERSONNEL DIVISION
Brooks Air Force Base, Texas 78235-5601

Reviewed by

Joseph L. Weeks, Chief
Cognitive Skills Assessment Branch

Submitted for publication by

William E. Alley, Technical Director
Manpower and Personnel Division

This publication is primarily a working paper. It is published solely to document work performed.

SUMMARY

The research presented here investigates the acquisition of Pascal programming skills in relation to individual differences in problem-solving abilities, incoming knowledge and skills, and learning style. Brooks (1977) proposed a framework of problem-solving abilities hypothesized to underlie programming skills (i.e., understanding, method-finding, and coding). These problem-solving abilities were estimated from performance on a battery of algebra (word problems) tests designed to map onto those specific problem-solving abilities. Prior knowledge and cognitive skills were assessed from scores on standardized tests covering broad areas of mastery (e.g., vocabulary knowledge), and a computerized battery of tests focusing on working memory capacity and information processing speed. Learning style was measured by the amount of assistance (i.e., hints) requested from the tutor--an active style being associated with requesting fewer hints. Finally, learning outcome or skill acquisition was estimated from performance on criterion tests measuring the breadth and depth of programming knowledge and skills acquired. The results indicate that learning Pascal programming is a function of Brooks' problem-solving abilities, working memory capacity, and an active learning style. This research suggests improvements in programming instruction focusing on those variables impacting programming skill acquisition or for the prediction of who will acquire programming skills for selection and classification purposes.

PREFACE

Development of this paper was supported by the Air Force Learning Abilities Measurement Program (LAMP), a multi-year program of basic research conducted at the Air Force Human Resources Laboratory and sponsored by the Air Force Office of Scientific Research. The goals of the program are to specify the basic parameters of learning ability, to develop techniques for the assessment of individuals' knowledge and skill levels, and to explore the feasibility of a model-based system of psychological assessment.

Support was provided by the Air Force Human Resources Laboratory and the Air Force Office of Scientific Research. We thank Dan Woltz, Pat Kyllonen, Bill Tirre, and Wes Regian for their valuable comments on this report.

TABLE OF CONTENTS

LIST OF TABLES

ACQUISITION OF PROGRAMMING SKILLS

I. INTRODUCTION

What are the characteristics of individuals who acquire programming skills efficiently and effectively? Can we successfully predict who is more likely to pick up programming skills from a computer programming curriculum? Can we improve the design of effective computer programming curricula? The research discussed in this paper investigated the relationship between programming skill acquisition and various measures of individual differences, including: (a) problem solving abilities (e.g., ability to decompose a problem into its constituent parts); (b) prior knowledge and cognitive skills (e.g., arithmetic reasoning, word knowledge, information processing speed); and (c) active versus passive learning style. Determining the prerequisite or correlated knowledge, skills, abilities, and traits for computer programming may ultimately provide educators with an explicit framework on which to base instruction.

Brooks (1977) investigated the characteristics of computer programmers. He presented a theoretical framework of the problem solving abilities involved in computer programming, postulating three major proficiencies or abilities that need to be active during programming: understanding, method-finding, and coding. According to Brooks (1977), the first problem solving ability or stage in the solution of a programming problem involves *understanding*, which refers to the identification of basic elements in a problem. Included in this stage is the determination of the properties and relations of problem elements, establishing the initial and final problem states, and hypothesizing the operations needed for achieving the problem solution. The second problem solving ability or stage of processing, *method-finding*, involves the decomposition and sequencing of the problem elements into an outline of the problem solution. That is, what are the relevant operators or commands and how should they be arranged in the solution of a programming problem? The final stage, *coding*, is the process of translating the natural language solution (from the previous stages) into programming code. Although this model has not been extensively tested, other researchers have found support for the processes of understanding (e.g., Faries & Reiser, 1988), method-finding (Mayer, Dyck, & Vilberg, 1986; Snow, 1980; and Swan & Black, 1987) and coding (e.g., Mayer et al., 1986).

Data measuring prior knowledge and skills in this study were obtained from scores on the Armed Services Vocational Aptitude Battery (ASVAB), Department of Defense (1984) covering a broad range of knowledge and skills such as vocabulary knowledge and mathematical skills. Cognitive process measures, more fundamental skills, were obtained from scores on a battery of computerized tests developed in the Learning Abilities Measurement Program (LAMP) at the Air Force Human Resources Laboratory (AFHRL), gauging working memory capacity and information processing speed in each of three domains: quantitative, verbal and spatial.

One goal of this research was to verify the existence and explicate the nature of the relationship between the three abilities hypothesized by Brooks and programming skill acquisition from an intelligent tutoring system instructing Pascal programming. The approach employed to study these relationships involved creating and administering an algebra word problem test battery, where the individual tests were constructed to parallel the problem solving abilities hypothesized by Brooks[1] as important factors in programming. Another goal of the research was to ascertain other contributory factors and variables that may be related to efficient and effective programming skill acquisition (e.g., working memory capacity, quantitative abilities, and so on).

[1]Brooks' second postulated ability of "method-finding" was subdivided into the two subcomponents: (a) decomposition of problem parts, and (b) sequencing parts into an outlined solution.

II. METHOD

Subjects

The subjects in this study consisted of 265 males and females participating in a 7-day study on acquisition of Pascal programming skills from an intelligent tutoring system (Shute, 1989a). Subjects were recruited and selected from local colleges and technical schools to match demographic and general ability characteristics of the Air Force enlisted population. Approximately 80% of the sample were males and 20% females. All subjects were high school graduates (or equivalent) with a mean age of 22.4. None of the subjects had any prior Pascal programming experience. All subjects were paid for their participation consisting of 40 hours of testing and learning.

Materials

The Pascal programming intelligent tutoring system (Pascal ITS) used in this study was originally developed at the Learning Research and Development Center, University of Pittsburgh (Bonar, Cunningham, Beatty, & Weil, 1988) and extensively modified at AFHRL. This system runs on a Xerox 1186 computer and was designed to help nonprogrammers learn how to program in Pascal.

The curriculum consisted of 25 programming problems of increasing difficulty. There were three learning phases per problem, and each phase was designed to teach different skills associated with programming. In phase 1, subjects generated natural language solutions to various programming problems. In phase 2, subjects selected and sequenced "programming plans" in the form of a flowchart for each problem. In phase 3, subjects translated the phase 2 visual solutions into Pascal code (see Bonar et al., 1988, for a complete description of the system). Within each problem, subjects were required to proceed sequentially through the three phases and they could not go on to the next phase until they had succeeded in the current phase. Subjects could take as long as they needed to go through the entire curriculum and could ask for an unlimited number of hints from the system for help in problem solution. Up to 30 hours were allowed for interaction, and no subject required more than that time. The mean time spent on the tutor was 12.2 hours, with a 5.2 hour standard deviation. The minimum time to complete the entire ITS was 2.8 hours and the maximum time was 29.2 hours (N = 260). These data were normally distributed.

A criterion posttest battery was created and administered online that tested the breadth and depth of knowledge and skills acquired from the Pascal ITS. This battery consisted of three tests, each one requiring progressively more complex programming skills. The first test involved the detection of errors in simple Pascal programs. Subjects were given a problem statement, presented with some Pascal code, and were asked to determine if an error was present, and if so, what type of error it was (e.g., unnecessary line, misplaced line, missing line). Subjects then had to identify the part of the program (the line) which manifested the error. The second test involved decomposing and ordering Pascal commands into a problem solution. Here, subjects received a problem statement, then a menu of possible commands (e.g., Readln, Write, If...Then...). They selected and arranged the commands to solve the programming problem in an adjacent window. In the third test, subjects generated and wrote Pascal code from scratch in response to programming problem statements. There were 12 problems per test, isomorphic to the tutor's problems, and subjects could take as long as they needed to complete the tests. Accuracies per problem per test as well as response latencies were automatically tallied and recorded by the system. Scoring of the third test (i.e., writing Pascal code) was done offline, by two scorers, and interscorer reliability was high (r = .90).

The algebra word problems test battery consisted of four increasingly difficult tests measuring the following specific problem solving abilities: (a) problem type identification, (b) problem decomposition into relevant arithmetic operators, (c) sequencing of the operators, and (d) translation of verbal problem statements into algebraic equations. As stated earlier, this test battery was created to map onto the abilities proposed by Brooks (1977) as fundamental to learning to program.

Mapping between Brooks' hypothesized processes and the algebra tests is direct: understanding parallels problem identification in that the goal of the understanding phase is to identify what type of problem needs to be solved. Method-finding maps onto decomposing and sequencing steps in a problem because it results in a program outline where each problem step is explicitly identified and placed in a particular order. Finally, coding involves the translation of natural language solution statements into programming code, comparable to the translation of arithmetic operators into an algebraic equation. This paper and pencil test took approximately 30 minutes for subjects to complete and consisted of 10 problems per section.

The ASVAB was included in this study to see how incoming knowledge and skills related to the acquisition of a new skill (i.e., Pascal programming). The ASVAB consists of 10 separate tests: General Science (GS), Arithmetic Reasoning (AR), Word Knowledge (WK), Paragraph Comprehension (PC), Numerical Operations (NO), Coding Speed (CS), Auto Shop (AS), Mechanical Comprehension (MC), Math Knowledge (MK), and Electronics Information (EI). This paper and pencil battery took 3.5 hours to complete and is used by the various armed services for selection and classification of enlisted personnel. All subjects in this research were administered the ASVAB prior to the Pascal ITS.

The LAMP at AFHRL conducts basic research concerning individual differences in cognitive abilities and skill acquisition. As part of this program, different computerized tests, administered on Zenith 248 microcomputers, have been developed during the past 5 years that measure the cognitive attributes of working memory (WM) capacity and information processing speed (IPS) in each of three different domains: quantitative, verbal, and spatial (Kyllonen & Christal, in press). Working memory tests require an individual to engage in concurrent storage and processing. That is, a WM test requires a person to store some information in memory while simultaneously processing new information. The degree to which individuals can handle this dual tasking without becoming overloaded reflects their WM capacity. Processing speed tests require an individual to answer various items as quickly as possible without sacrificing accuracy.

A computerized test battery was developed for this study (see Appendix A) involving three computerized tests in each of these six categories: WM Quantitative, WM Verbal, WM Spatial, IPS Quantitative, IPS Verbal, and IPS Spatial. In addition, we had a composite measure of general knowledge (GK) Verbal.

Procedure

Subjects were tested in groups of approximately 14 persons, and there were 20 groups tested all together. Each group spent 7 days (nearly 6 hours per day) in this research. Subjects began the study being tested on the basic cognitive process measures. Over successive days, they were administered the ASVAB, followed by the algebra word problems test battery, the ITS (up to 30 hours), and the criterion test battery.

III. RESULTS

The first research question concerned the relationship between problem solving abilities, as measured by the algebra word problems tests, and success in learning Pascal, as measured by

The first research question concerned the relationship between problem solving abilities, as measured by the algebra word problems tests, and success in learning Pascal, as measured by scores on the criterion tests. The simple correlations between the algebra test scores and the criterion test scores (as well as summary statistics) can be seen in Table 1. The first observation from this table is that the algebra test data, representing problem solving abilities, and criterion performance were significantly correlated. Thus, there is evidence supporting Brooks' model relating particular problem solving abilities to programming skill acquisition. Indeed, all of the hypothesized abilities (measured by the corresponding algebra tests) correlated with success in acquiring various programming skills as represented and required by each criterion test in the battery.

Table 1 **Percent Correct, Standard Deviations and Correlations Among Algebra Tests and Criterion Tests**

			Correlations			
Test	**Mean**	**SD**	**Crit1[a]**	**Crit2**	**Crit3**	**Crit**
Problem identification	65.9	18.3	.41*	.42*	.52*	.50*
Decomposition	82.2	18.4	.48*	.51*	.55*	.57*
Sequencing	68.2	25.9	.55*	.55*	.61*	.64*
Translation	60.1	22.0	.48*	.48*	.55*	.56*
Overall[b]	69.3	17.7	.58*	.59*	.67*	.69*
Crit1	38.4	23.3		.64*	.74*	
Crit2	76.1	15.7			.71*	
Crit3	52.3	24.5				
Crit	55.8	19.0				

Note $N = 257$

[a]Crit1 is percent correct on part 1 of the criterion test battery, Crit2 corresponds to percent correct on part 2, and Crit3 is the percent correct from part 3 of the test battery. Crit is the mean score of all three criterion tests combined

[b]Overall represents the mean score from all four of the algebra tests.

*$p < .001$

The correlation between the mean scores from the algebra and criterion test batteries was $r = .69$, $p < .001$. Thus, there is a highly significant correlation existing between general levels of performance on these two tests (i.e., general problem solving skills and success in learning Pascal programming skills).

A regression analysis was performed on these data to see which of the problem solving abilities predicted final performance (i.e., the collapsed criterion score, "Crit"). Three of the four tests remained in the equation after backward elimination: (a) problem identification or understanding, (b) sequencing, and (c) translation or coding. The high multiple $R = .70$ ($R^2 = .49$) indicates that these three tests or "problem solving abilities" are important in predicting successful programming skill acquisition for novice subjects.[2] So, almost half the variance of criterion performance can be explained by these three abilities.

Another goal of this research was to examine the relationship involving other variables, such as incoming knowledge and skills, in predicting the acquisition of Pascal programming skills. Correlations among ASVAB test data and our final performance measures (i.e., individual criterion

[2]The test measuring decomposition skills was eliminated because its simple correlation to sequencing was very high ($r = .90$) thus it did not contribute any new variance to the equation.

tests and collapsed accuracy score) revealed that almost all of the individual ASVAB tests significantly correlated to some degree with our criterion scores. So, not only are certain problem solving abilities like problem identification and sequencing important to programming skill acquisition, but having a broad, general knowledge base is also important to learning programming skills (e.g., math and word knowledge). Table 2 shows the summary statistics of the ASVAB data from our sample as well as correlations with our final performance measures.

Table 2. **Percent Correct, Standard Deviations and Correlations Among ASVAB Data and Criterion Test Scores**

				Correlations			
Test	Abbrev.	Mean	SD	Crit1	Crit2	Crit3	Crit
General Science	GS	52.8	8.8	.50**	.45**	.53**	.55**
Arithmetic Reasoning	AR	52.1	9.2	.52**	.51**	.58**	.60**
Word Knowledge	WK	53.0	7.0	.39**	.42**	.48**	.48**
Paragraph Comprehension	PC	52.5	7.5	.35**	.29**	.40**	.40**
Numerical Operations	NO	52.8	8.4	.22**	.14	.22**	.23**
Coding Speed	CS	52.1	8.0	.28**	.21**	.27**	.29**
Auto Shop	AS	53.1	8.9	.17*	.18*	.21**	.21**
Math Knowledge	MK	52.5	9.0	.59**	.50**	.61**	.64**
Mechanical Comprehension	MC	53.0	9.9	.46**	.47**	.49**	.53**
Electronics Information	EI	52.9	9.2	.36**	.37**	.45**	.44**

Note. N = 260.

*p < .01.

**p < .001.

Because the number of individual tests involved in both the ASVAB and LAMP batteries are collectively too large for enable regression analyses, we decided to derive and use composite measures for both the ASVAB and LAMP data in subsequent analyses. Standard composite scores are available from the ASVAB measuring knowledge and skills for electronics, mechanics, administration and general knowledge, in addition to general aptitude (i.e., Armed Forces Qualification Test). However, these composites are typically used by the Air Force for classification purposes and they are not mutually exclusive. That is, the same test can be included in more than one composite.

Factor analyses of the 10 ASVAB tests conducted by Ree, Mullins, Mathews and Massey (1982) as well as Fairbank, Tirre and Anderson (in press) on very large samples have consistently found four factors: Verbal Ability, Quantitative Ability, Perceptual Speed, and Technical Knowledge. The associated ASVAB tests loading the most on these factors are as follows: Verbal Ability: PC, WK, GS; Quantitative Ability: AR, MK; Perceptual Speed: NO, CS; and Technical Knowledge: EI, AS, MC. For our sample, we standardized each individual's ASVAB test scores, then created our own four composite measures averaging the relevant test scores per factor. Intercorrelations among the composite values and the criterion test scores are displayed in Table 3.

The main feature of these correlations is that the quantitative and, to a lesser extent, the verbal ability factors are the most highly correlated with our criterion measures, and also with each other. Technical knowledge, while correlating with the verbal and quantitative factors, is not as associated with the final outcome measures. A backward elimination regression analysis was performed on these data predicting overall accuracy on the criterion test battery from the four ASVAB composites. Two composites remained in the final equation (multiple R = .68; R^2 = .46): Quantitative Ability (t = 8.7; p < .001), and Verbal Ability (t = 3.1; $p \leq$ .01).

Table 3. **Correlations Among ASVAB Composites and Criterion Tests**

	Crit1	Crit2	Crit3
Verbal	.48**	.45**	.54**
Quant	.60**	.54**	.64**
Speed	.28**	.20**	.28**
Tech	.37**	.39**	.43**
	Quant	**Speed**	**Tech**
Verbal	.66**	.19*	.68**
Quant		.43**	.60**
Speed			.15*

Note. N = 257.
*p < .01.
**p < .001.

Next, we created composite scores from the LAMP cognitive abilities tests measuring working memory (WM) capacity and information processing speed (IPS) in each of three domains: verbal, quantitative, and spatial. For each of the six categories of cognitive attribute by domain, there were three tests given for reliability purposes. In other words, we used three tests measuring verbal WM, three tests measuring spatial IPS, and so on. When collapsed across domain, there were nine tests per composite measure (WM and IPS). We used these higher level composites for our analyses. Additionally we had a measure of General Knowledge (GK) in the verbal domain, itself a composite of four tests.

A backward elimination regression analysis was performed on the LAMP composite data predicting overall accuracy on the criterion test battery. Two composites remained in the final equation (multiple R = .68; R^2 = .46): Working Memory (t = 8.7; p < .001) and General Knowledge/Verbal (t = 3.6; p < .001). Therefore, larger working memory capacity, in conjunction with general verbal knowledge predict successful learning of Pascal programming.

To coalesce these separate findings, all three classes of predictor variables were included in a regression analysis. Also included in this analysis was a variable, ACTIV, a learning style measure of activity (i.e., passive to active). This variable, a simple count of how many hints the person asked for from the ITS (where fewer hints were associated with being more active) has been shown to be strongly correlated with performance both in terms of rate of learning, and learning outcome (see Shute, 1989b, for a more detailed discussion of this topic).

Regression analyses of these data were performed testing full and restricted models of the data. The 12 main variables,[3] 12 squared main effects, and 21 two-way interactions involving WM and the activity index were tested (full model) predicting the composite criterion (mean accuracy on the criterion test battery). This resulted in a multiple R = .86; adjusted R^2 = .67. Next, we performed a backward elimination of the interactions and squared variables. Only one interaction (ACTIV and WM) and one squared term (ACTIV squared) remained in the equation (multiple R = .84). Finally, a regression analysis with backward elimination was performed of the main effects. Results of this analysis (see Table 4) yielded a model with four main effects, an interaction, and a squared term in the equation (multiple R = .84, R^2 = .70; N = 250).

[3]The 12 variables include the four problem solving abilities from the algebra test data, the four ASVAB composites, the three LAMP composites, and the learning style measure (activity).

Table 4. **Results of Regression Analyses**

Variable	Beta	T	Significance T	Unique Variance
ACTIV	-.90	-7.1	.0000	6.9%
WM	.31	4.4	.0000	2.6%
ACTIV/SQ	.51	4.1	.0001	2.3%
SEQ	.20	3.9	.0001	2.1%
IDEN	.12	2.8	.0060	1.1%
ACTIV/WM	-.13	-2.0	.0466	0.6%

The results were interpreted to mean that 70% of the variance of the dependent measure can be accounted for by a learning style variable (ACTIV), a cognitive process measure (WM), and two problem solving abilities (IDEN and SEQ).

First, the learning style index (i.e., linear and quadratic functions) accounted for over 9% of the unique variance of our learning criterion. The more hints one asked for, the less knowledge and skills were acquired by the end of the tutor (inverse linear function), particularly after initial familiarization with the system where asking for hints was an acceptable behavior in terms of orienting to a novel learning environment.

Second, the LAMP working memory (WM) composite also proved to be a good predictor of our criterion, accounting for 2.6% of the unique variance of our criterion measure. In this case, the larger an individual's WM capacity, the more he or she was capable of learning from the Pascal programming ITS. Moreover, the significant interaction involving WM and number of hints requested suggested that, although having low WM capacity can be a major hindrance for learning programming skills, an even worse condition occurs when one has low WM capacity in conjunction with the trait of asking for many hints.

Third, two of our problem solving ability measures entered into the final equation: sequencing problem elements (SEQ) and problem identification or understanding (IDEN). These two variables together accounted for over 3% of the unique variance in our criterion. Because of the high correlation between decomposition and sequencing (r = .90), decomposition failed to contribute any unique variance to the equation and thus was not included in the final result.

One more regression analysis was performed to see how much additional variance in the criterion measure could be explained by variables beyond the contribution made by the ASVAB composite data once the ASVAB data were entered first and allowed to fit the data optimally. We entered the four composites into the regression analysis, then stepped in the other variables. The ASVAB composites alone accounted for 47% of the variance. Adding the other variables into the equation (i.e., those variables that were significant from the full and restricted model testing discussed above: ACTIV, WM, SEQ, ID, ACTIV/Squared and ACTIV/WM), the R^2 increased from .47 to .70, thus explaining an additional 23% of the criterion variance.

IV. DISCUSSION

The question asked at the beginning of this paper concerned the characteristics of individuals who successfully acquired programming skills. This issue can be addressed in terms of the knowledge, skills, abilities and traits tested in this research. Regarding the validity of the model proposed by Brooks (i.e., the problem solving abilities of understanding, method-finding, and coding as requisites to programming skill acquisition), the results indicate that, indeed, the proposed general abilities all significantly correlate with successful programming skill acquisition.

In particular, the general processes of understanding problems and sequencing problem elements are the most predictive of successful learning of Pascal programming from our intelligent tutoring system than Brooks' translation/coding or decomposition abilities. One explanation for this may be that in our sample of novice programmers, the most difficult conceptual hurdle for the majority of them had to do with formulating and organizing an outline solution for a given problem. Once that structure was established (i.e., sequenced appropriately), translating it into Pascal code was relatively easy since the tutor provided extensive support in the translation phase of learning. The pedagogical approach taken by the ITS emphasized the higher, conceptual level of programming (i.e., organizing a solution) more than the lower level syntactical aspects of Pascal programming. Thus, these findings based on the relative importance of the processes of understanding and sequencing are not surprising in light of the design and implementation of the ITS.

Another goal of this research was to investigate the relationship between incoming knowledge and skills and programming skill acquisition. First, correlations involving ASVAB and criterion data (Table 2) show that a wide range of general knowledge tests (e.g., science, math, word, mechanical, electronics) are highly correlated with successful learning of Pascal programming skills. Individuals with more general knowledge across different domains learn Pascal better, as evidenced by higher scores on their criterion test battery. One possible reason for this may be that persons with more diverse knowledge have more efficient storage, retention, and/or retrieval mechanisms, or have more intrinsic motivation for learning (i.e., more intellectual curiosity). Alternatively, individuals scoring well on the ASVAB and well on the criterion posttest battery may simply be good test-takers. The more prior knowledge one can apply to a new learning situation, the more one can learn by analogy. So, new learning may be limited or mediated by prior knowledge as well as WM capacity (how much more one is capable of learning).

The composites from the LAMP battery measuring WM and general verbal knowledge were also shown to be significantly correlated to and predictive of programming skill acquisition thus supporting the above proposition concerning the knowledge and capacity limitations imposed on new learning.

Some of the findings may be explained as a function of the learning environment itself. For instance, the learning task used in this study (i.e., the Pascal ITS) presented learners with an initial problem statement which they had to understand and then attempt to structure a solution, from informal natural language to more formal Pascal code. An example problem statement for one of the latter, more difficult problems was:

> Write a program that prompts the user for the balance of an employee payment account. Then ask the user to enter the income for individual employees. Continue asking for an employee's income until the sum of the employees' incomes is greater than or equal to the employee payment account. Report the number of employees that you were able to pay out of this account.

There are two salient observations in regard to this problem statement. First, a great deal of initial parsing must be conducted on this statement in order to understand what is being requested (i.e., good verbal skills are required to comprehend the question). Second, understanding the problem statement also relies on WM to maintain important aspects of the problem as a whole. In support of this hypothesized interaction among problem understanding and both general verbal knowledge and WM capacity, we found that "understanding," as measured from the first algebra test of the battery, not only was a good predictor of programming skill acquisition from our tutor, but was also significantly correlated with GKV and WM (.39 and .47, respectively).

Other findings are purely a function of the learner. A particular learning style indicator, hint-asking, showed a highly significant negative correlation with learning outcome: Figuring out solutions on one's own without excessive assistance from a tutor ultimately leads to more learning than a more passive style of requesting help from an intelligent tutor. When the 25 Pascal programming problems were divided into fifths (i.e., first five problems, second five, and so on), to see changes in behavior over time, the pattern of correlations between number of hints requested and accuracy on the criterion test battery showed the following trend. For the earliest five problems, where the correlation between hint-asking and overall criterion posttest accuracy was $\underline{r} = -.47$, it is not as detrimental to seek assistance as with the second through fifth groups (correlations with overall accuracy = -.48, -.56, -.66, -.66, respectively). So, in the latter problems, persistent hint-asking adversely affects the amount and kind of knowledge and skills acquired from the tutor, but not so much in the initial, orienting stages of learning (note also the significant quadratic trend: ACTIV-Squared supporting the "later hint-asking is worse than earlier hint-asking" tenet). So, an individual's activity level during the learning process may be a function of either learning style (e.g., lethargy) or knowledge deficit (e.g., does not know something, therefore asks for hints). Regardless of the basis for this behavior, a possible remedy may be to either impose a limitation on the number of hints a person may receive, and/or reinforce an individual's active participation in the learning process.

The overall picture that emerges from these findings is that success in learning Pascal programming is a function of good problem solving abilities, WM capacity, and being an active learner. What are the implications of these findings for programming teachers and ITS developers? Since certain problem solving abilities, as delineated by Brooks and tested in this research, are highly correlated with successful acquisition of programming skills, and since these same abilities are believed to be trainable (e.g., part-task training of sequencing skills), computer programming curricula may benefit from the inclusion of supplemental instruction on relevant problem solving skills.

Information about an individual's cognitive process measures may be used to vary instruction in a principled manner, such as teaching smaller chunks of relevant knowledge for those with lower WM capacity. In other words, since WM capacity was shown to be an important predictor of programming skill acquisition, and the functional size of an individual's WM cannot be directly manipulated, instruction may be varied based on differing units of knowledge for those individuals determined (by simple testing) to have smaller WM capacities.

In conclusion, a very large amount of the criterion variance (70%) can be explained by just a few variables. We can use this information to enhance instruction, focusing on those variables impacting programming skill acquisition, or we can use the findings to predict who will acquire good programming skills for selection and classification purposes. If the designated variables can be instructed and/or trained, we can maximize instruction for more individuals, which is the purpose of ITS's, in particular, and education, in general. For purposes of selection and classification, our findings highlight the variables that allow us to predict who is likely to acquire programming skills.

REFERENCES

Bonar, J.G., Cunningham, R., Beatty, P., & Weil, W. (1988). *Bridge: Intelligent tutoring with intermediate representations* (Technical Report). LRDC, University of Pittsburgh, Pittsburgh, PA.

Brooks, R. (1977). Toward a theory of the cognitive process in computer programming. *International Journal of Man-Machine Studies, 9*, 737-751.

Department of Defense. (1984). *Test manual for the Armed Services Vocational Aptitude Battery* (DOD, 1340.12AA). North Chicago, IL: U.S. Military Entrance Processing Command.

Fairbank, B.A., Jr., Tirre, W.C., & Anderson, N.S. (in press). Measures of thirty cognitive tasks: Analysis of reliabilities, intercorrelations, and correlations with aptitude battery scores. To appear in P.L. Dann (Ed.), *Advances in computerized testing*. Dordiecht, the Netherlands: Martinns Nijhoff.

Faries, J.M., & Reiser, B.J. (1988, June). *Access and use of previous solutions in a problem solving situation* (CSL Technical Report 29). Princeton University, Princeton, NJ.

Kyllonen, P.C., & Christal, R.E. (in press). Cognitive modeling of learning abilities: A status report of LAMP. In R.Dillon & J.W. Pellegrino (Eds.), *Testing: Theoretical and applied issues*. San Francisco: Freeman.

Mayer, R.E., Dyck, J.L., & Vilberg, W. (1986). Learning to program and learning to think: What's the connection? *Communications of the ACM, 29(7)*, 605-610.

Ree, M.J., Mullins, C.J., Mathews, J.J., & Massey, R.H. (1982). *Armed Services Vocational Aptitude Battery, Item and factor analyses of Forms 8, 9, and 10* (AFHRL-TR-81-55, AD A113 465). Brooks Air Force Base, TX: Manpower and Personnel Division, Air Force Human Resources Laboratory.

Shute, V.J. (1989a). Learning abilities and intelligent tutoring systems. In *Proceedings from the Technology and Innovations in Training and Education conference* (TITE '89), Atlanta, GA.

Shute, V.J. (1989b). Individual differences in learning from an intelligent tutoring system. *Technology and Learning, 3*(3), 7-11.

Snow, R.E. (1980). Aptitude processes. *Aptitude, Learning and Instruction, 1*, 27-63.

Swan, K., & Black, J.B. (1987). *The cross-contextual transfer of problem solving skills*. New York: Teachers College, Columbia University.

APPENDIX A: BATTERY OF LAMP TESTS -- CAM Ver. 1.0
(Even-Odd Reliabilities in Parentheses)

I. WORKING MEMORY (Percent correct)

A. Quantitative

ABC Recall: Subjects must learn and remember numeric values assigned to the letters A, B, and C. Statements (e.g., A = B/2) are presented one at a time, and subjects are permitted to look at each one for as long as desired before going on to the next statement. They are then asked to recall the values of the letters one at a time. Some of the problems are more difficult than others in that the values must be computed (e.g., A = 2 x 8 thus, A = 16). Still other values may not be able to be computed until the value of another letter is known (e.g., B = A + 4). (.95)

Mental Math: This task requires subjects to calculate a subtraction or division problem mentally, and then choose the correct answer from 5 alternatives. A problem appears on the screen for 2 seconds (preceded by a warning asterisk) and then disappears. Subjects mentally solve the problem for as long as they wish. When they have the answer, they hit the space bar to see the five alternative solutions. They have 4 seconds to type in the number of the correct answer. (.88)

Slots Test: This test presents simple math equations (e.g., 5 + 2) in five sequential positions on the screen. Subjects must calculate the equations as they are presented and remember the answer for each position. Following the presentation of equation stimuli, a question mark appears in one of the positions, and subjects must type in the corresponding answer. The five positions are marked by horizontal lines, one next to the other. Problems are presented from left to right, one at a time. Two rates of presentation exist (i.e., slow and fast) and before each trial, subjects are warned to get ready for either a slow or fast item. Each problem presents between 1 and 10 math equations. In the more difficult items, new math problems may be presented in a slot where a problem was already presented. Subjects are required to remember the most recent answer. Subjects are told whether their response was correct. (.91)

B. Verbal

ABCD Test: Subjects are presented with five general rules:

Rule 1 - Set 1 = A and B.
Rule 2 - Set 2 = C and D.
Rule 3 - Set 1 can either precede or follow Set 2.
Rule 4 - A can either precede or follow B.
Rule 5 - C can either precede or follow D.

Each problem consists of three instructions presented one at a time concerning Set 1 and 2. *For example:* (a) A precedes B, (b) Set 1 follows Set 2, and (c) C precedes D. Each instruction is presented one at a time, and subjects may look at each one as long as desired before going on to the next one. They must determine the appropriate order of the letters and then hit the space bar to see the choices of answers. They then choose the number for the correct answer (e.g., CDAB). (.81)

Dan's Word Span Test: Subjects are required to memorize a short list of words and answer questions about them. A "Get Ready!!" warning precedes the words, which are presented one at a time. The questions are asked in a math problem format. For example, if the list were "neat, burp, inn," a possible question is "neat + 1 = ?." This question asks for the word which is one position after "neat." Another question could be "inn - 2 = ?." This question asks for the word residing two positions before "inn." Answers are presented in a multiple choice format; alternatives are synonyms to the actual words on the list. Subjects must type in the number for the synonym which matches the word from the list. Any given word list is between 3 to 5 words long. Subjects answer three questions about each list, after which they are told how many questions they had correct for that word list. (.93)

Reading Span: (Danneman & Carpenter's task). This task tests subjects' ability to classify true/false statements and their short-term memory capacity. Subjects are presented a list of sentences of general knowledge which they must determine to be true/like ("L") or false/different ("D"). Concurrently, they must memorize the last word in each sentence (this word is highlighted a different color from the other words). Sentences are presented one at a time, after which they are asked to type in the first two letters of each word in the order that they appeared. Subjects receive partial credit if the correct letters are typed in, but in the wrong sequence. The number of sentences in a list begins with 2 and ends with 6 for classification and word recall. (.93)

C. Spatial

Figure Synthesis: Two geometric figures are presented for subjects who are instructed to imagine the shape if the pieces were rearranged to form one figure. These figures are then replaced by a third figure. The subject must determine whether or not the third figure could be formed from the combined figures. Reaction time is presented when subjects give the correct response. (.62)

Spatial Visualization: (Guilford's Spatial Visualization task). This task tests the ability to work problems requiring 3-dimensional visualization. Subjects read descriptions of blocks and visualize how they appear before and after various manipulations (e.g., colors, initial size, ensuing size, number of blocks it may be cut into, etc.). The subject is allowed to study the description for 30 seconds before the first question is asked (although the description remains on the screen throughout the problem). Subjects work the problems mentally and then choose one of the multiple choice answers using the letters A through O. Subjects are given 60 seconds to respond to a question, at which time they are told to enter their response (within another 10 seconds). If no response is entered during that time, the item is counted wrong. For each description, three or more questions may be asked in this multiple choice format. (.84)

Ichikawa: This test presents a 5x5 matrix of squares in which 7 contain asterisks. The placement of the asterisks is randomly determined, so as not to allow an easily memorized configuration. Subjects see a warning asterisk, the matrix filled with asterisks, and then a blank matrix with a question mark in one of the squares. Subjects are to determine whether or not an asterisk was in that square, and respond with "L" (correct) or 'D' (not correct). Subjects have 3 seconds to respond, and then a new blank matrix appears with another question mark in it. For each matrix, three positions are questioned. The computer tells subjects if their response was correct or wrong. Subjects are allowed to study the initial matrix for 2 seconds, followed by a 1 second delay before questions are asked. (.73)

II. INFORMATION PROCESSING SPEED (Latencies)

A. Quantitative

Number Fact Reduction: Subjects are presented four sets of simple arithmetic problems, each set containing only one type of problem (i.e., addition, subtraction, multiplication, or division). Each problem is preceded by a "Get Ready!!" warning and asterisk. Subjects must quickly determine whether or not the problem is correct (type in "L") or incorrect (type in "D"). The computer will respond with a "correct" message with the subject's reaction time or a "wrong" message. (.98)

Moyer Landaur: Subjects are presented with two single-digit numbers on separate sides of the screen to determine which of the two is larger. If the one on the right is greater, "L" is the correct response; and if the one on the left is greater, "D" is the correct response. Each set of numbers is preceded by a warning asterisk and a one second delay. The test contains four sets of 36 trials. (.99)

Odd Even Test: In this test subjects must decide as quickly as possible whether two numbers presented are odd or even. The two numbers (between 1 and 20) are presented one above the other. Some numbers are presented as digits and others as English words (e.g., 5 or five). Subjects will respond with "L" if both numbers are either odd or even If one number is odd and the other is even, then "D" is the correct response. Reaction time is shown when a response is entered. (.97)

B. Verbal

Meaning Identity: Two words are presented and the subject must decide whether they have the same or different meanings. Subjects are to type in "L" if they have the same meaning, and "D" if they have different meanings. Some of the pairs of words are repeated and some words are repeated with a different pair. Each pair is preceded by a warning asterisk. The goal is to respond as quickly as possible and still try to get 95% of the items correct. After each set, the student's percent correct and average response time are shown. (.98)

Category Identification: Subjects are presented with three words: one on the left, one on the right, and one centered above the other two. The subjects must determine which of the lower words belongs in the same class or category as the word at the top. If it is the one on the left, "D" is the correct response, and "L" if it is the one on the right. Three warning asterisks are presented where the words will appear for the subjects to focus their attention. The computer responds with whether their answer is correct or incorrect in addition to reporting their reaction time. (.98)

Semantic Relations Verification Test: Subjects must determine whether or not simple sentences are true ("L") or false ("D"). Key words in the sentence are colored; for example, in the sentence "Theft is a crime," "theft" and "crime" might be colored differently from the default color of white like the rest of the sentence. The computer responds with whether or not the subject made the correct response, and the reaction time if the response is correct. (.98)

C. Spatial

Santa's Figures: Subjects are presented with 2 sets of geometric figures to determine if they have the same parts ("L") or different parts ("D"). Each set consists of three geometric shapes or figures (e.g., circle, arrow, diamond, square), one next to the other. The first set

appears for 2 seconds, disappears, and then the second set appears. Subjects are required to make their decisions. The order of the figures is not important, only whether or not the two sets contain the same figures. (.93)

Palmer's Figure Comparison: Subjects are presented with two 3x3 dot-matrices with 5 interconnecting lines forming different geometric patterns. These matrices are presented side by side until a response is entered. The subject must respond as quickly as possible whether the two line figures are the same ("L") or different ("D"). The computer responds with either "correct" or "wrong"; and at the end of each set it tells the student how many responses were correct. (.97)

D-String Matching: This task presents two strings of letters, symbols, or digits for subjects to determine if they are the same ("L") or different ("D"). The strings may be upper or lower case letters, numbers, or any other symbol from the keyboard (e.g., * & % $). Each string is between 2 to 5 characters long. The two stimuli comprising a comparison are always the same length and composed of the same character type (e.g., upper case with upper case, symbols with symbols, etc.). Strings may be made up of the exact characters, but if the sequence is different, then it is not a match. In all cases, the two strings will either be identical, transposed, or just one value may be different. Subjects are presented a warning asterisk and then the two strings. The strings stay on the screen until a response is entered. At that time, the computer responds with whether the response was correct and the reaction time. (.97)

II. GENERAL KNOWLEDGE (Percent correct)

A. Verbal

General Knowledge Survey: Subjects are asked general questions (e.g., San Antonio is in what state?), and must respond by typing in the first two letters of the answer (e.g., "TE" for Texas). The computer responds with "correct" or "wrong." (.93)

Reading Span True/False (Danneman & Carpenter's task). This task tests subjects' ability to classify statements as either true or false. Subjects are presented a list of sentences of general knowledge which they must determine to be true/like ("L") or false/different ("D"). (.70)

Meaning Identity: Meaning ID. Two words are presented and the subject must decide whether they have the same or different meaning. Subjects are to type in "L" if they have the same meaning, and "D" if they have different meanings. (.80)

Semantic Relations Verification Test: Subjects must determine whether or not simple sentences are true ("L") or false ("D") (e.g., "Theft is a crime" would be a true sentence). The computer responds with whether or not the subject made the correct response, and the reaction time if the response is correct. (.84)